**河南开封科技传媒学院实验报告**

••••••••••••••••••••••••••••••••• 密 ••••••••••••••••••••••••••••••••• 封 ••••••••••••••••••••••••••••••••• 线 •••••••••••••••••••••••••••••••••

**20 -20 学年第 学期**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 姓名： | 刘东 | | 学号： | 2136101045 | 专业： | 软件工程 | |
| 所在学院： | | 理工学院 | | | 指导老师： | | 袁帅 |
| 实验时间： | | 2023/3/29 | | | 实验地点： | | 6503 |
| 课程名称： | | 数据结构 | | | | | |
| 实验题目： | | 第六周实验 | | | | | |
| 实验目的： | | | | | | | |
| 实验内容：  字符串的常规操作  字符串的左旋转操作是把字符串前面的若干个字符转移到字符串的尾部。请定义一个函数实现字符串左旋转操作的功能。比如，输入字符串"abcdefg"和数字2，该函数将返回左旋转两位得到的结果"cdefgab"。  示例 1：  输入: s = "abcdefg", k = 2  输出: "cdefgab"  示例 2：  输入: s = "lrloseumgh", k = 6  输出: "umghlrlose"  限制：  1 <= k < s.length <= 10000  思考题：KMP算法的相关应用  实现strStr()函数。  给你两个字符串haystack和needle，请你在haystack字符串中找出needle字符串出现的第一个位置（下标从0开始）。如果不存在，则返回-1。  说明：  当needle是空字符串时，我们应当返回什么值呢？这是一个在面试中很好的问题。  对于本题而言，当needle是空字符串时我们应当返回0。这与C语言的strstr()以及Java 的indexOf()定义相符。  示例 1：  输入：haystack="hello",needle ="ll"  输出：2  示例 2：  输入：haystack="aaaaa",needle="bba"  输出：-1  示例 3：  输入：haystack = "", needle = ""  输出：0  提示：  0 <= haystack.length, needle.length <= 5 \* 104  haystack 和 needle 仅由小写英文字符组成 | | | | | | | |
| 实验代码：  #include<bits/stdc++.h>  using namespace std;  int main() {  string str;  string temp;  int k;    cin >> str;  cin >> k;a  int i = 0;  int left = 0, right = k;  while (right < str.size() || left < k) {  if (left < k) temp += str[left++];  if (right < str.size()) str[i++] = str[right++];  }  left = 0;  for (; i < str.size(); i++) str[i] = temp[left++];  cout << str << endl;      return 0;  }  思考：  #include<bits/stdc++.h>  using namespace std;  void getNext(int\* next, const string& s) {  next[0] = 0;  int i = 1, j = 0;  while (i <= s.size()) {  if (j == 0 || s[i] == s[j]) next[i++] = j++;  else j = next[j];  }  }  int strStr(string haystack, string needle) {  if (needle.empty()) {  return 0;  }  int next[needle.size()];  getNext(next, needle);  int j = 0;  for (int i = 0; i < haystack.size(); i++) {  while(j > 0 && haystack[i] != needle[j]) {  j = next[j - 1];  }  if (haystack[i] == needle[j]) {  j++;  }  if (j == needle.size() ) {  return (i - needle.size() + 1);  }  }  return -1;  }  int main() {  string haystack, needle;  cin >> haystack >> needle;    cout << strStr(haystack, needle);    return 0;  } | | | | | | | |
| 实验结果截图及实验心得： | | | | | | | |